

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Examiner Search

Date: 9/14/04

CiteSeer

Find: incremental updating signal

Documents

Citations

Searching for PHRASE incremental updating signal.

Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) CSB

DBLP

No documents match Boolean query. Trying non-Boolean relevance query.

500 documents found. Order: relevance to query.

WebML: Querying the World-Wide Web for Resources and Knowledge - Zaiane, Han (1998) (Correct) (10 citations)
has been initially conceived. The updates are incremental and are propagated, in the case of the schema after the system has been initially conceived. The updates are incremental and are propagated, in the case
ftp.fas.sfu.ca/pub/cs/han/kdd/webq98.ps

Approximate Processing and Incremental Refinement... - Winograd, Ludwig (1995) (Correct)
July 1995: Approximate Processing and Incremental Refinement Concepts J. Winograd J.
refinement can be implemented from the following update equations. The ith successive approximation, X_i
a systematic tradeoff between the quality of signal processing results and the availability of
rassp.scrz.org/public/conts/2ndrefine.ps

Reinforcement - Learning In (Correct)
require a model to be given or learned. It is incremental, requiring only a constant amount of
Learning In Continuous Time: Advantage Updating Leemon C. Baird lli.bairdic@wi.wpaib.af.mil
the controller is given a scalar reinforcement signal (or cost function) indicating how well it is
www.cs.cmu.edu/~leemon/papers/cnn94/cnn94.ps

Incremental Methods for Formal Verification and Logic Synthesis - Swamy (1996) (Correct) (3 citations)
Incremental Methods for Formal Verification and Logic
www-cad.eecs.berkeley.edu/~gms/pub/gms/PhDThesis.ps.Z

Notes on the Computation of the Incremental Gain of Linear... - Submitted To (Correct)
Notes on the Computation of the Incremental Gain of Linear Systems with Saturation
discrete time plants (even over some bounded signal spaces for which the induced norm exists) and
throughout is the Euclidean norm, denoted $\|x\|$. The signal spaces used are all p spaces (where k, p
www-control.eng.cam.ac.uk/bgr/bgr_cd95s.ps

Incremental Updating of Objects in INDIGO - Bouzy (1997) (Correct)
Incremental Updating of Objects in INDIGO Bruno Bouzy
Incremental Updating of Objects in INDIGO Bruno Bouzy LAFORIA-IBP
www.math-info.univ-paris5.fr/~bouzy/publications/incremental_article.ps.Z

Software Environment For The Development Of Embedded Signal... - Joseph Winograd (Correct)
is described. The environment encourages incremental design via modular and hierarchical
Environment For The Development Of Embedded Signal Processing Systems Joseph M. Winograd And S.
environment for the rapid development of embedded signal processing software is described. The
ftp.cs.umass.edu/pub/less/winograd-icassp95-c++-ps

Variations on Incremental Interpretation - Stuart Shieber (1993) (Correct) (6 citations)
Variations on Incremental Interpretation Stuart Shieber Mark Johnson
connectionist networks has assumed synchronous updating of activation levels, many researchers have
the other extreme would be a single reset or error signal to the syntactic processor to stop processing
www.eecs.harvard.edu/~shieber/papers/incremental-interp.ps.gz

Yan Lin Marek J. Druzdzal - Decision Systems Laboratory (Correct)
updating. It is desirable in such systems to incrementally update beliefs rather than recomputing the
can be used to improve efficiency of belief updating algorithms by identifying and pruning those
www.isp.pitt.edu/~yan/myPapers/fairs98.ps

Modeling RF Circuits for Systems Analysis - Chen, Phillipps, Silveira, Kundert (1998) (Correct)
as a ratio of output to input, not the ratio of incremental output to incremental input. The key

http://citeseer.ist.psu.edu/cis?q=incremental+updating+signal&cs=1

9/9/04

http://citeseer.ist.psu.edu/cis?q=incremental+updating+signal&cs=1

9/9/04

region of operation the K-model simulates large-signal base-band transients to within about 10% of those
RF carriers. Complex variables represent RF signals. The real part represents the amplitude of the
aligos.inesc.pt/~lms/publications/cic98-interp.ps.gz

When to Update the Sequential Patterns of Stream Data - Zheng, Xu, Ma (2002) (Correct)
of sequential patterns. The experiments for the incremental updating algorithm IUS on two data sets show
www.nisde.buaa.edu.cn/~kexu/papers/wupdate.pdf

Constructing Multi-View Editing Environments Using MViews - Grundy, Hosking (1993) (Correct) (2 citations)
after one view is updated. This should support incremental view updates and visual indication of changes
so all views are made consistent after one view is updated. This should support incremental view updates
ftp.cs.waikato.ac.nz/pub/papers/mviews/VL93.ps.gz

Properties of Languages That Make Recursive Views... - Dong, Libkin, Wong (Correct)
of this paper show that in most cases such incremental maintenance is impossible if either no
functions. Finally, we relate the complexity of updating transitive closure to that of updating the
sdmc.krdi.org.sg/leis/ips/Zdwi97-7.ps

Generating Fast QR Algorithms Using Signal Flow Graph Techniques - Marc Moonen (Correct)
matrix R, together with a vector z are stored and updated, such that at any point in time the optimal
Generating Fast QR Algorithms Using Signal Flow Graph Techniques Marc Moonen Katholieke
Maivern Worcesterhire, VR14 3PS, UK prouder@signal.dra.hmg.go Abstract in this paper we will show
ftp.esat.kuleuven.ac.be/pub/SISTA/moonen/reports/asilomar96.ps.Z

Incremental Cryptography and Application to Virus... - Bellare, Goldreich (1995) (Correct) (15 citations)
Incremental Cryptography and Application to Virus
theory.cs.mit.edu/pub/people/oded/bgg-inc2.ps

Incremental Algorithms on Lists - Jeuring (1991) (Correct) (2 citations)
Incremental algorithms on lists Johan Jeuring CWI P.O.
www.cs.chalmers.se/~johan/formatting.ps

Systematic Derivation of Incremental Programs - Yanhong Liu (1995) (Correct) (14 citations)
Systematic Derivation of Incremental Programs Yanhong A. Liu y Tim
ftp.cs.indiana.edu/pub/lu/liu-inc-SCP95.ps

Parallel Incremental Scheduling - Wu (1995) (Correct) (3 citations)
If World Scientific Publishing Company Parallel Incremental Scheduling Min-You Wu Department of Computer
ftp.cs.buffalo.edu/pub/tech-reports/95-46.ps.Z

Deriving Incremental Programs - Liu, Teitelbaum (1993) (Correct)
Deriving Incremental Programs Yanhong A. Liu Tim Teitelbaum
ftp.cs.indiana.edu/pub/lu/Drv-TR93-Rvs.ps.Z

New Methods For Surface Reconstruction From Range Images - Brian Lee Curless (1997) (Correct) (7 citations)
set of properties for such algorithms includes: incremental updating, representation of directional
for such algorithms includes: incremental updating, representation of directional uncertainty, the
.46 3.5 A Signal Processing Perspective
graphics.stanford.edu/pub/curless/thesis/thesis.ps.gz

First 20 documents Next 20

Try your query at: Google (CiteSeer) Google (Web) CSB DBLP

CiteSeer.IST - Copyright NEC and ISI

CiteSeer Find: incremental updating message

Searching for PHRASE Incremental updating message.

Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) CSB
 DRIP

১৪৮

**No documents match Boolean query. Trying non-Boolean relevance query.
500 documents found. Order: relevance to query.**

WebML: Querying the World-Wide Web for Resources and Knowledge - Zaiane, Han (1998) (Correct) (10 citations)
has been initially conceived. The updates are Incremental and are propagated, in the case of the schema after the system has been initially conceived. The updates are Incremental and are propagated, in the case
ftp.fas.slu.se/pub/cs/han/kdd/webq98.ps

Incremental Methods for Formal Verification and Logic Synthesis - Swamy (1996) (Correct) (3 citations)

Incremental Methods for Formal Verification and Logic
www-cad.eecs.berkeley.edu/~gms/pub/gmsPhdThesis.ps Z

Incremental Updating of Objects in INDIGO - Bouzy (1997) (Correct)

Incremental Updating of Objects in INDIGO Bruno Bouzy

Incremental Updating of Objects in INDIGO Bruno Bouzy LAFORIA-IBP
www.math-info.univ-paris5.fr/~bouzy/publications/incremental.article.ps.Z

Global Causal Ordering with Minimal Latency - Koch, Moser, Melliar-Smith (Correct)

	Sends a further message m 2 to the group, updating the initialized item.	Receives initial state	Initial model	(Context)
R	sends a further message m 2 to the group, updating the initialized item.	Delivery of m 2 to S		

Abstract Causally order delivery of messages is a useful property for coordinating the
of replicated information. Timestamps in the messages can be used to impose a causal order on the
beta.ece.ucsb.edu/~ruppert/ps/causal.ps.gz

Yan Lin Marek J. Druzdel - Decision Systems Laboratory (Correct)

updating. It is desirable in such systems to incrementally update beliefs rather than recomputing the entire belief set. In this paper, we propose a new algorithm for belief updating that can be used to improve efficiency of belief updating algorithms by identifying and pruning those beliefs that are not needed for future updates. The algorithm is implemented in a system called *Belief Pruning*. The system is evaluated using a set of benchmark problems. The results show that the algorithm can significantly reduce the number of beliefs that need to be updated, thereby improving the efficiency of belief updating algorithms.

An algebraic semantics of Message Sequence Charts - Mauw, Reniers (1994) (Correct) (6 citations)

An algebraic semantics of **Message Sequence Charts** S. Mauw M.A. Reniers Dept. of
spoke@win.tue.nl michel@win.tue.nl **Abstract** **Message Sequence Charts** is a graphical and textual
former CTT) The main area of application for **Message Sequence Charts** is as an overview specification
ftp.win.tue.nl/pub/techreports/michel/ComputingScienceNotes94-23.ps.z

Implementation and Performance of Transparent, - Elnozahy, Zwaenepoel (Correct)

the overhead during failure-free performance. Incremental and copy-on-write checkpointing is used to checkpointing, and iii) support for atomic updates. Unlike conventional file servers that are with checkpointing and volatile sender-based message logging to simultaneously achieve low

ftp.cs.ucsd.edu/pub/imports/manetho1.ps.Z

Supporting Timing-Channel Free Computations In Multi-level... - Sandhu, Thomas, Jajodia (1991) (Correct) (1 citation)
flows. This is because a restricted method cannot update the state (attributes) of an object whereas an earlier paper [3]Jajodia and Kogan proposed a message filter approach to enforcing mandatory security object-oriented databases. The key idea in the message filter model is that all information exchange be
www.lqtmu.edu.cn/conf/mcf/tp/as_ver/9100.ps

When to Update the Sequential Patterns of Stream Data - Zheng, Xu, Ma (2002) (Correct)
of sequential patterns. The experiments for the incremental updating algorithm IUS on two data sets show
www.nlsde.buaa.edu.cn/~kexu/papers/wupdate.pdf

Constructing Multi-View Editing Environments Using MViews - Grundy, Hosking (1993) (Correct) (2 citations)

<http://citeseer.ist.psu.edu/cis?q=incremental+updating+message&cs=1>

9/9/04

<http://citeseer.ist.psu.edu/cis?q=incremental+updating+message&cs=1>

flp.cs.waikato.ac.nz/pub/papers/mviews/L93.ps.gz

Properties of Languages That Make Recursive Views. - Dong, Libkin, Wong (Correct)
of this paper show that in most cases such incremental maintenance is impossible if either no
functions. Finally, we relate the complexity of updating transitive closure to that of updating the
sdmc.krdi.org.sa/kleis/psZ/dlw/c97-7.ps

Incremental Cryptography and Application to Virus.. - Bellare, Goldreich.. (1995) (Correct) (15 citations)

Incremental Algorithms on Lists - Jeuring (1991) (Correct) (2 citations)
Incremental algorithms on lists Johan Jeuring CWI P.O.
www.cs.cwi.nl/~johan/formatting.ps

Systematic Derivation of Incremental Programs - Yanhong Liu (1995) (Correct) (14 citations)
 Systematic Derivation of Incremental Programs Yanhong A. Liu y Tim
[ftp.cs.indiana.edu/pub/iliu/inc-SCP95.ps](http://cs.indiana.edu/pub/iliu/inc-SCP95.ps)

Mutable Objects - Pierce (1993) (Correct) (1 citation)

function takes a class and a description of an incremental change to its state and behavior, and builds a new object that represents the state and behavior after the change. The function returns the new object.

Parallel Incremental Scheduling - Wu (1995) (Correct) (3 citations)

of Wu Department of Computer Science, National Tsing-Tung University, Hsinchu, Taiwan. E-mail: mls@cs.nthu.edu.tw

A Simple Approach to Parallel Functional Programming - Roe (Correct)

Ironically these programs require destructive updating of arrays, which traditionally is difficult to functionalize. On top of this low level message passing model we aim to build some general functional programs. The first problem by working with a very simple message passing model of parallelism, which is easy to

A First Implementation of Modular Smalltalk - Holst, Szafron (1993) (Correct)

Method Dispatch Based On New Extensions To Incremental Cache Table Coloring. Key Words: Modular
 11183, 11184, 11185, 11186, 11187, 11188, 11189, 11190, 11191, 11192, 11193, 11194, 11195, 11196, 11197, 11198, 11199, 11200, 11201, 11202, 11203, 11204, 11205, 11206, 11207, 11208, 11209, 11210, 11211, 11212, 11213, 11214, 11215, 11216, 11217, 11218, 11219, 11220, 11221, 11222, 11223, 11224, 11225, 11226, 11227, 11228, 11229, 11230, 11231, 11232, 11233, 11234, 11235, 11236, 11237, 11238, 11239, 11240, 11241, 11242, 11243, 11244, 11245, 11246, 11247, 11248, 11249, 11250, 11251, 11252, 11253, 11254, 11255, 11256, 11257, 11258, 11259, 11260, 11261, 11262, 11263, 11264, 11265, 11266, 11267, 11268, 11269, 11270, 11271, 11272, 11273, 11274, 11275, 11276, 11277, 11278, 11279, 11280, 11281, 11282, 11283, 11284, 11285, 11286, 11287, 11288, 11289, 11290, 11291, 11292, 11293, 11294, 11295, 11296, 11297, 11298, 11299, 11300, 11301, 11302, 11303, 11304, 11305, 11306, 11307, 11308, 11309, 11310, 11311, 11312, 11313, 11314, 11315, 11316, 11317, 11318, 11319, 11320, 11321, 11322, 11323, 11324, 11325, 11326, 11327, 11328, 11329, 11330, 11331, 11332, 11333, 11334, 11335, 11336, 11337, 11338, 11339, 11340, 11341, 11342, 11343, 11344, 11345, 11346, 11347, 11348, 11349, 11350, 11351, 11352, 11353, 11354, 11355, 11356, 11357, 11358, 11359, 11360, 11361, 11362, 11363, 11364, 11365, 11366, 11367, 11368, 11369, 11370, 11371, 11372, 11373, 11374, 11375, 11376, 11377, 11378, 11379, 11380, 11381, 11382, 11383, 11384, 11385, 11386, 11387, 11388, 11389, 11390, 11391, 11392, 11393, 11394, 11395, 11396, 11397, 11398, 11399, 11400, 11401, 11402, 11403, 11404, 11405, 11406, 11407, 11408, 11409, 11410, 11411, 11412, 11413, 11414, 11415, 11416, 11417, 11418, 11419, 11420, 11421, 11422, 11423, 11424, 11425, 11426, 11427, 11428, 11429, 11430, 11431, 11432, 11433, 11434, 11435, 11436, 11437, 11438, 11439, 11440, 11441, 11442, 11443, 11444, 11445, 11446, 11447, 11448, 11449, 11450, 11451, 11452, 11453, 11454, 11455, 11456, 11457, 11458, 11459, 11460, 11461, 11462, 11463, 11464, 11465, 11466, 11467, 11468, 11469, 11470, 11471, 11472, 11473, 11474, 11475, 11476, 11477, 11478, 11479, 11480, 11481, 11482, 11483, 11484, 11485, 11486, 11487, 11488, 11489, 11490, 11491, 11492, 11493, 11494, 11495, 11496, 11497, 11498, 11499, 11500, 11501, 11502, 11503, 11504, 11505, 11506, 11507, 11508, 11509, 11510, 11511, 11512, 11513, 11514, 11515, 11516, 11517, 11518, 11519, 11520, 11521, 11522, 11523, 11524, 11525, 11526, 11527, 11528, 11529, 11530, 11531, 11532, 11533, 11534, 11535, 11536, 11537, 11538, 11539, 11540, 11541, 11542, 11543, 11544, 11545, 11546, 11547, 11548, 11549, 11550, 11551, 11552, 11553, 11554, 11555, 11556, 11557, 11558, 11559, 11560, 11561, 11562, 11563, 11564, 11565, 11566, 11567, 11568, 11569, 11570, 11571, 11572, 11573, 11574, 11575, 11576, 11577, 11578, 11579, 11580, 11581, 11582, 11583, 11584, 11585, 11586, 11587, 11588, 11589, 11590, 11591, 11592, 11593, 11594, 11595, 11596, 11597, 11598, 11599, 11600, 11601, 11602, 11603, 11604, 11605, 11606, 11607, 11608, 11609, 11610, 11611, 11612, 11613, 11614, 11615, 11616, 11617, 11618, 11619, 11620, 11621, 11622, 11623, 11624, 11625, 11626, 11627, 11628, 11629, 11630, 11631, 11632, 11633, 11634, 11635, 11636, 11637, 11638, 11639, 11640, 11641, 11642, 11643, 11644, 11645, 11646, 11647, 11648, 11649, 11650, 11651, 11652, 11653, 11654, 11655, 11656, 11657, 11658, 11659, 11660, 11661, 11662, 11663, 11664, 11665, 11666, 11667, 11668, 11669, 11670, 11671, 11672, 11673, 11674, 11675, 11676, 11677, 11678, 11679, 11680, 11681, 11682, 11683, 11684, 11685, 11686, 11687, 11688, 11689, 11690, 11691, 11692, 11693, 11694, 11695, 11696, 11697, 11698, 11699, 11700, 11701, 11702, 11703, 11704, 11705, 11706, 11707, 11708, 11709, 11710, 11711, 11712, 11713, 11714, 11715, 11716, 11717, 11718, 11719, 11720, 11721, 11722, 11723, 11724, 11725, 11726, 11727, 11728, 11729, 11730, 11731, 11732, 11733, 11734, 11735, 11736, 11737, 11738, 11739, 11740, 11741, 11742, 11743, 11744, 11745, 11746, 11747, 11748, 11749, 11750, 11751, 11752, 11753, 11754, 11755, 11756, 11757, 11758, 11759, 11760, 11761, 11762, 11763, 11764

Deriving Incremental Programs - Liu, Teitelbaum (1993) (Correct)

Deriving Incremental Programs Yanhong A. Liu Tim Teitelbaum
<http://www.cse.indiana.edu/pub/iliu/Dry-TR93-Rys.ps.Z>

New Methods For Surface Reconstruction From Range Images - Brian Lee Curless (1997) (Correct)
set of properties for such algorithms includes: Incremental updating, representation of directional
for such algorithms includes: Incremental updating, representation of directional uncertainty, the
graphics.stanford.edu/pub/curless/thesis/thesis.ps.gz

First 20 documents **Next 20**

Try your query at: [Google \(CiteSeer\)](#) [Google Web](#) [CSB](#) [DBLP](#)

CiteSeer.IST - Copyright NEC and IST

9/9/04

CiteSeer

Find:

DocumentsCitations

Searching for PHRASE revision signal.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)
9 documents found. Order: number of citations.

[Working T13 Draft 1532D Volume 1 - Revision March June \(Correct\)](#)
Item 3 adding note about slew rate on RESET signal Revision 1 -August 28, 2002 Added the following
[www.t13.org/docs2003/d1532v1r3.pdf](#)

[Working T13 Draft 1532D Volume 3 - Revision April March \(2003\) \(Correct\)](#)
Item 3 adding note about slew rate on RESET signal Revision 1 -August 28, 2002 Added the following
[www.t13.org/docs2003/e03116r2.pdf](#)

[Working T13 - Draft Volume Revision \(Correct\)](#)
Item 3 adding note about slew rate on RESET signal Revision 1 -August 28, 2002 Added the following
[www.t13.org/docs2003/d1532v3r2.pdf](#)

[Unknown - Draft Volume Revision \(Correct\)](#)
Item 3 adding note about slew rate on RESET signal Revision 1 -August 28, 2002 Added the following
[www.t13.org/docs2003/d1532v3r2a.pdf](#)

Try your query at: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

[CiteSeer](#) - Copyright [NEC](#) and [IST](#)

CiteSeer Find: incremental revision signal

Searching for PHRASE incremental revision signal.
 Restrict to: **Header** **Title** Order by: **Expected citations** **Hubs** **Usage** **Date** Try: **Google** **(CiteSeer)** **Google (Web)** **CSB** **DBLP**

No documents match Boolean query. Trying non-Boolean relevance query.
 500 documents found. Only retrieving 250 documents (System busy - maximum reduced). Order: relevance to query.

FRST - An Interactive Revision System for Forward Chaining Rule... - Feldman (1994) (Correct) (1 citation)
 performed. In this paper we present a system for incremental revision of rule bases called FRST (Forward
 1 FRST - An Interactive Revision System for Forward Chaining Rule Bases Ronen
 ftp://ri.frlrli/conferences/KA-ML-94/feldman.ps.gz

Indexing, Elaboration and Refinement: Incremental Learning of... - Ram (1993) (Correct)
 indexing, elaboration and refinement: incremental learning of explanatory cases Ashwin Ram
 a theory of incremental learning based on the revision of previously existing case knowledge in
 ftp://cc.gatech.edu/pub/ai/ram/fit-cc-92-03.ps.z

Using Case-Based Reasoning to Acquire User Scheduling... - Sycara, Zeng, Miyashita (1995) (Correct)
 change over time. This paper advocates an incremental revision framework for improving schedule
 over time. This paper advocates an incremental revision framework for improving schedule quality and
 www.cs.cmu.edu/~sycara/cabins-papers/caia95-final.ps.gz

Approximate Processing and Incremental Refinement... - Winograd, Ludwig.. (1995) (Correct)
 July 1995 Approximate Processing and Incremental Refinement Concepts J. Winograd J.
 a systematic tradeoff between the quality of signal processing results and the availability of
 filtering. 2) real-time time-frequency analysis of signals and (3) DCT-based image encoding/decoding.
 rassp.scra.org/public/conts/2nd/refine.ps

CABINS: A Framework of Knowledge Acquisition and Iterative... - Miyashita, Sycara (1995) (Correct) (13 citations)
 preferences and tradeoffs and using them to incrementally improve schedule quality in predictive
 Framework of Knowledge Acquisition and Iterative Revision for Schedule Improvement and Reactive Repair
 www.cs.cmu.edu/~sycara/cabins-papers/aj-cabins-final.ps.gz

Notes on the Computation of the Incremental Gain of Linear... - Submitted To (Correct)
 Notes on the Computation of the Incremental Gain of Linear Systems with Saturation
 discrete time plants (even over some bounded signal spaces for which the induced norm exists) and
 throughout is the Euclidean norm, denoted $\|x\|$. The signal spaces used are all p spaces (where $k \leq p$
 www-control.eng.cam.ac.uk/bgr/bgr_cdc95a.ps

Software Environment For The Development Of Embedded Signal... - Joseph Winograd (Correct)
 is described. The environment encourages incremental design via modular and hierarchical
 Environment For The Development Of Embedded Signal Processing Systems Joseph M. Winograd And S.
 environment for the rapid development of embedded signal processing software is described. The
 ftp://cs.umass.edu/pub/lesser/winograd-icassp95-c++.ps

Variations on Incremental Interpretation - Stuart Shieber (1993) (Correct) (6 citations)
 Variations on Incremental Interpretation Stuart Shieber Mark Johnson
 the other extreme would be a single reset or error signal to the syntactic processor to stop processing
 manner. Under a loose interaction model, the reset signal can occur before the end of the sentence. 2.1
 www.eecs.harvard.edu/~shieber/papers/incremental-interp.ps.gz

Multi-Agent Belief Revision - Kfir-Dahav, Tennenholtz (Correct)
 Multi-Agent Belief Revision Noa E. Kfir-Dahav and Moshe Tennenholtz Faculty
 lew3.technion.ac.il:8080/~moshet/mabr.ps

Embedding Revision Programs in Logic Programming Situation Calculus - Baral (1997) (Correct) (1 citation)
 Embedding Revision Programs in Logic Programming Situation
 cs.utep.edu/chitta/papers/rule-based-updates-journal.ps

http://citeseer.ist.psu.edu/cis?q=incremental+revision+signal&cs=1

9/9/04

http://citeseer.ist.psu.edu/cis?q=incremental+revision+signal&cs=1

9/9/04

Incremental Methods for Formal Verification and Logic Synthesis - Swamy (1996) (Correct) (3 citations)
 Incremental Methods for Formal Verification and Logic
 www-cad.eecs.berkeley.edu/~gms/pub/gmsPhdThesis.ps.z

A Unified View of Belief Revision and Update - Val, Shoham (1994) (Correct) (7 citations)
 A Unified View of Belief Revision and Update Alvaro del Val and Yoav Shoham
 robotics.Stanford.EDU/~shoham/bel_change.ps

Modeling RF Circuits for Systems Analysis - Chen, Phillips, Silveira, Kundert (1998) (Correct)
 as a ratio of output to input, not the ratio of incremental output to incremental input. The key
 region of operation the K-model simulates large-signal base-band transients to within about 10% of those
 RF carriers. Complex variables represent RF signals. The real part represents the amplitude of the
 algos.inesc.pt/~lms/publications/cfcs88-interp.ps.gz

Belief Revision - Segal (1994) (Correct)
 an ATMS. The conflicts are used one at a time to incrementally build the set of minimum candidates. The
 Belief Revision Richard Segal Department of Computer Science and
 www.cs.washington.edu/homes/segal/belief-revision.ps.gz

Semantics of First Order Belief Revision Based on Circumscription - Li Yan (1992) (Correct)
 Semantics of First Order Belief Revision Based on Circumscription Li Yan Yuan and
 menaik.cs.ualberta.ca/pub/TechReports/1992/TR92-17/TR92-17.ps

Incremental Algorithms on Lists - Jeuring (1991) (Correct) (2 citations)
 Incremental algorithms on lists Johan Jeuring CWI P.O.
 www.cs.chalmers.se/~johan/formatting.ps

Systematic Derivation of Incremental Programs - Yanhong Liu (1995) (Correct) (14 citations)
 Systematic Derivation of Incremental Programs Yanhong A. Liu y Tim
 ftp://cs.indiana.edu/pub/liu/linc-SCP95.ps

Mutual Belief Revision - van der Meyden (1994) (Correct) (2 citations)
 Mutual Belief Revision (Preliminary Report) Ron van der Meyden NTT
 www-staff.socs.uts.edu.au/8090/~ron/research/mbr.ps

Parallel Incremental Scheduling - Wu (1995) (Correct) (3 citations)
 fl World Scientific Publishing Company Parallel Incremental Scheduling Min-You Wu Department of Computer
 ftp://cs.buffalo.edu/pub/tech-reports/95-46.ps.z

Deriving Incremental Programs - Liu, Teitelbaum (1993) (Correct)
 Deriving Incremental Programs Yanhong A. Liu Tim Teitelbaum
 ftp://cs.indiana.edu/pub/liu/Drv-TR93-Rvs.ps.z

First 20 documents Next 20

Try your query at: **Google** **(CiteSeer)** **Google (Web)** **CSB** **DBLP**

CiteSeer - Copyright NEC and IST

CiteSeerFind:

Searching for PHRASE incremental software update.

Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) CSB DBLPNo documents match Boolean query. Trying non-Boolean relevance query.
500 documents found. Only retrieving 250 documents (System busy - maximum reduced). Order: relevance to query.

Using Write Protected Data Structures To Improve Software Fault... - Sullivan (1991) (Correct) (9 citations)
 Using Write Protected Data Structures To Improve Software Fault Tolerance in Highly Available Database
 Temporarily unprotect the data structures during updates. The paper discusses the effects of three
 The paper discusses the effects of three different update models on performance, software complexity, and
 s2k-fp.cs.berkeley.edu:8000/postgres/papers/ERL-M91-56.ps.Z

Incremental Recomputation of Active Relational Expressions - Qian (1991) (Correct) (84 citations)
 ENGINEERING, VOL. 3, NO. 3, SEPTEMBER 1991 337
 Incremental Recomputation of Active Relational
 database systems that support engineering or software environments[56]it is essential to
 Qian and Gio Wiederhold Abstract-Database updates are small and Incremental compared to database
 www.csl.sri.com/~qian/tkde91.ps.gz

Experiments With Destructive Updates in a Lazy Functional... - Hartel, Vree (1994) (Correct) (1 citation)
 arrays may be a good alternative to using the Incremental approach with destructive updates. Keywords:
 functional languages using flow graph analysis. Software-practice and experience, 24(2)127-173, Feb
 Experiments with destructive updates in a lazy functional language Pieter H. Hartel
 www.wins.uva.nl/pub/functional/reports/CL_update.ps.Z

David G. Goodenough - Daniel Charlebois (Correct)
 forest inventory update. SEIDAM uses a set of software agents that carry out tasks such as translate
 Automated Forest Inventory Update with SEIDAM David G. Goodenough 1.4 Daniel
 about the process of digital forest inventory update. SEIDAM uses a set of software agents that carry
 www.engr.uvic.ca/~ndaley/nigelfinv_update.ps

System Support for Software Fault Tolerance in Highly Available... - Sullivan (1992) (Correct) (5 citations)
 System Support for Software Fault Tolerance in Highly Available Database
 wuarchive.wustl.edu/packages/postgres/papers/ERL-M93-05.ps.Z

Constructing Multi-View Editing Environments Using MViews - Grundy, Hosking (1993) (Correct) (2 citations)
 after one view is updated. This should support incremental view updates and visual indication of changes
 could be used to support distributed multi-user software development. Acknowledgments The financial
 so all views are made consistent after one view is updated. This should support incremental view updates
 ftp.cs.waikato.ac.nz/pub/papers/mviews/VL93.ps.gz

A Review of Post-Factum Software Integration Methods - Isazadeh, MacEwen, Malton (1995) (Correct)
 the original system's functionality. 3. Incremental Engineering: A software system can be
 A Review of Post-Factum Software Integration Methods Ayaz Isazadeh Glenn H.
 www.qcids.queensu.ca/Department/TechReports/Reports/1995-389.ps

SQL Can Maintain Polynomial-Hierarchy Queries - Libkin, Wong (1997) (Correct)
 1 Summary Let us introduce the concept of an incremental evaluation system, or IES, discussed in [15]
 optional auxiliary database, and a finite set of "update" functions that correspond to the different kinds
 correspond to the different kinds of permissible update to the input database. These update functions
 sdmc.krdl.org.sg/kleis/psZ/lw-ph-23sept97.ps

Approximate Processing and Incremental Refinement - Winograd, Ludwig... (1995) (Correct)
 July 1995 Approximate Processing and Incremental Refinement Concepts J. Winograd J.
 refinement can be implemented from the following update equations. The i th successive approximation, X_i
 $k \in \{ \Gamma \Gamma \Gamma \Gamma \Gamma \}$ where $C_i(k)$ is the coverage update, which is defined as $C_i(k) \Delta \Gamma \Gamma \Gamma \Gamma \Gamma$
 rassp.sora.org/public/conts/2nd/refine.ps

http://citeseer.ist.psu.edu/cis?q=incremental+software+update&cs=1

9/9/04 http://citeseer.ist.psu.edu/cis?q=incremental+software+update&cs=1

9/9/04

Deterministic Semantics of Set-Oriented Update Sequences - Laasch, Scholl (1993) (Correct) (7 citations)
 Deterministic Semantics of Set-Oriented Update Sequences Christian Laasch, Marc H. Scholl
 is proposed that allows to apply sequences of update operations in a set-oriented way with
 triggers that are applied after (set-oriented) SQL updates can be checked. Furthermore, the iterator can be
 ftp.fmi.uni-konstanz.de/pub/dbis/Publications/LS:ICDE93.ps.gz
 Incremental Methods for Formal Verification and Logic Synthesis - Swamy (1996) (Correct) (3 citations)
 Incremental Methods for Formal Verification and Logic
 www-cad.eecs.berkeley.edu/~gms/pub/gmsPhdThesis.ps.Z

PORTAL Subscribe (Full Service) Register (Limited Service, Free) Login
 Search: ☐ The ACM Digital Library ☐ The Guide
 Incremental software update

THE ACM DIGITAL LIBRARY

Terms used **incremental software update** Found **34,227** of **141,680**

Sort results by: **relevance** Try an **Advanced Search**
 Display results: **expanded form** Try this search in **The ACM Guide**
☐ Open results in a new window

Results 1 - 20 of 200 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)
 Best 200 shown Relevance scale ☐ ☐ ☐ ☐

- 1 **An incremental algorithm for software analysis**
 Martin Carroll, Barbara G Ryder
 January 1987 **ACM SIGPLAN Notices , Proceedings of the second ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments**, Volume 22 Issue 1

Full text available: [pdf\(778.98 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In programming environments aimed at "industrial strength" software development, there is a need for software tools which facilitate both design and maintenance. These tools should encourage experimentation with different system configurations which enable designers to a priori estimate the associated system complexity and judge the ease of accommodating enhancements. Maintainers should be able to check straightforwardly the ramifications of system changes due to ...

- 2 **Full papers: Tree bitmap: hardware/software IP lookups with incremental updates**
 Will Eatherton, George Varghese, Zubin Dittia
 April 2004 **ACM SIGCOMM Computer Communication Review**, Volume 34 Issue 2

Full text available: [pdf\(189.39 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Even with the significant focus on IP address lookup in the published literature as well as focus on this market by commercial semiconductor vendors, there is still a challenge for router architects to find solutions that simultaneously meet 3 criteria: scaling in terms of lookup speeds as well as table sizes, the ability to perform high speed updates, and the ability to fit into the overall memory architecture of an Level 3 forwarding engine or packet processor with low systems cost overhead. 1 ...

- 3 **Incremental data-flow analysis algorithms**
 Barbara G. Ryder, Marvin C. Paull
 January 1988 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 10 Issue 1

Full text available: [pdf\(3.62 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

An incremental update algorithm modifies the solution of a problem that has been changed, rather than re-solving the entire problem. ACINCF and ACINCB are incremental update algorithms for forward and backward data-flow analysis, respectively, based on our equations model of Allen-Cocke interval analysis. In addition, we have studied their performance on a "nontoy" structured programming language L. Given a set of localized

<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27172042&CFTOKEN=69179037>
 Results (page 1): incremental software update

accurate than those obta ...

Keywords: Code optimization, Control flow, Data flow, Debugging, Internal program representation, Interpreter, Program slice, Software complexity metrics

- 8 **Linguistic support for the evolutionary design of software architectures**
 T. C. Nicholas Graham, Tore Urnes
 May 1998 **Proceedings of the 18th International conference on Software engineering**

Full text available: [pdf\(1.62 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Publisher Site](#)

As a program's functionality evolves over time, its software architecture should evolve as well so that it continues to match the program's design. This paper introduces the architecture language of Clock, a language for the development of Interactive, multiuser applications. This architecture language possesses three properties supporting the easy restructuring of software architectures: restricted scoping supported by a constraint-based communication system, automatic message routing, and easy ...

Keywords: Clock, Clock-Works programming environment, architecture language, automatic message routing, constraint-based communication, high level languages, Interactive systems, interactive, multiuser applications, programming environments, restricted scoping, software architectures, software engineering, vlsal syntax

- 9 **Compilers I: A framework for incremental extensible compiler construction**
 Steven Carroll, Constantine Polychronopoulos
 June 2003 **Proceedings of the 17th annual international conference on Supercomputing**

Full text available: [pdf\(182.20 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Much of the research in compiler design and optimization has traditionally focused on the effectiveness and efficiency of code optimization. However, the subject of efficiency of the entire compilation process itself (as opposed to the complexity of individual analysis or optimization algorithms) remains a highly complex and less investigated topic. In this paper we present a global approach to extensible and efficient compiler design, which aims at also improving the effectiveness and efficiency ...

Keywords: compilers, extensibility, Incremental analysis

- 10 **Matching-based incremental evaluators for hierarchical attribute grammar dialects**
 Alan Carle, Lori Pollock
 March 1995 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 17 Issue 2

Full text available: [pdf\(2.28 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Although attribute grammars have been very effective for defining individual modules of language translators, they have been rather ineffective for specifying large program-transformational systems. Recently, several new attribute grammar "dialects" have been developed that support the modular specification of these systems by allowing modules, each described by an attribute grammar, to be composed to form a complete system. Acceptance of these new hierarchical ...

Keywords: attribute grammar, hierarchical specifications, incremental evaluation, language translation, translators

<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27172042&CFTOKEN=69179037>

program changes in a program written in L, we identify ...

- 4 **Parallel and distributed incremental attribute evaluation algorithms for multiuser software development environments**

Gail E. Kaiser, Simon M. Kaplan
 January 1993 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 2 Issue 1

Full text available: [pdf\(3.09 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The problem of change propagation in multiuser software development environments distributed across a local-area network is addressed. The program is modeled as an attributed parse tree segmented among multiple user processes and changes are modeled as subtree replacements requested asynchronously by individual users. Change propagation is then implemented using decentralized incremental evaluation of an attribute grammar that defines the static semantic properties of the p ...

Keywords: attribute grammar, change propagation, distributed, Incremental algorithm, parallel, reliability

- 5 **Incremental analysis of side effects for C software system**

Jyh-Shian Yur, Barbara G. Ryder, William A. Landi, Phil Stocks
 May 1997 **Proceedings of the 19th international conference on Software engineering**

Full text available: [pdf\(1.90 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: dataflow analysis, Incremental analysis

- 6 **Incremental data flow analysis**

Barbara G. Ryder
 January 1983 **Proceedings of the 10th ACM SIGACT-SIGPLAN symposium on Principles of programming languages**

Full text available: [pdf\(865.20 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

In this paper we present ACINCF and ACINCB, incremental update algorithms for forward and backward data flow problems, which are based on a linear equations model of Allen/Cocke interval analysis [Allen 77, Ryder 82a]. We have studied their performance on a robust structured programming language L. Given a set of localized program changes in a program in L, we can identify *a priori* the nodes in its flow graph whose corresponding data flow equations will be affected by the ch ...

- 7 **The program dependence graph in a software development environment**

Karl J. Ottenstein, Linda M. Ottenstein
 April 1984 **Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments**, Volume 19, 9 Issue 5, 1

Full text available: [pdf\(942.88 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The internal program representation chosen for a software development environment plays a critical role in the nature of that environment. A form should facilitate implementation and contribute to the responsiveness of the environment to the user. The program dependence graph (PDG) may be a suitable internal form. It allows programs to be sliced in linear time for debugging and for use by language-directed editors. The slices obtained are more

9/9/04<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27172042&CFTOKEN=69179037>
 Page 3 of 6Results (page 1): incremental software update

9/9/04
 Page 4 of 6

- 11 **Direct update of data flow representations for a meaning-preserving program restructuring tool**

William G. Griswold
 December 1993 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1st ACM SIGSOFT symposium on Foundations of software engineering**, Volume 18 Issue 5

Full text available: [pdf\(1.64 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Automated assistance for meaning-preserving global restructuring is an approach for helping software engineers improve the structure of programs, thus lowering the costs of maintenance. The construction of a restructuring tool encounters many conflicting goals---such as simplicity, extensibility, and good performance---that cannot be met without some compromise. In particular, the current technique for assisting restructuring uses a costly program representation---a Program Dependence Graph (PDG) ...

- 12 **Software architecture based on communicating residential environments**

Erik Sandewall, Claes Strömberg, Henrik Sörensen
 March 1981 **Proceedings of the 5th international conference on Software engineering**

Full text available: [pdf\(864.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes an alternative approach to software architecture, where the classical division of responsibilities between operating systems, programming languages and compilers, and so forth is revised. Our alternative is organized as a set of self-contained environments which are able to communicate pieces of software between them, and whose internal structure is predominantly descriptive and declarative. The base structure within each environment (its divers ...

- 13 **Connecting software components with declarative glue**

Brian W. Beach
 June 1992 **Proceedings of the 14th international conference on Software engineering**

Full text available: [pdf\(1.33 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

- 14 **Incremental computation of complex object queries**

Hiroaki Nakamura
 October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**, Volume 36 Issue 11

Full text available: [pdf\(228.82 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The need for incremental algorithms for evaluating database queries is well known, but constructing algorithms that work on object-oriented databases (OODBs) has been difficult. The reason is that OODB query languages involve complex data types including composite objects and nested collections. As a result, existing algorithms have limitations in that the kinds of database updates are restricted, the operations found in many query languages are not supported, or the algorithms are too complex t ...

- 15 **Incremental cryptography and application to virus protection**

Mihir Bellare, Oded Goldreich, Shafi Goldwasser
 May 1995 **Proceedings of the twenty-seventh annual ACM symposium on Theory of computing**

Full text available: [pdf\(1.65 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

9/9/04<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27172042&CFTOKEN=69179037>





9/9/04

Full text available:  pdf(1.29 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**Keywords:** incremental analysis, interprocedural pointer aliasing, interprocedural side effect analysis**20 A software model and specification language for non-WIMP user interfaces**Robert J. K. Jacob, Leonidas Deligiannidis, Stephen Morrison
March 1999 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 6 Issue 1Full text available:  pdf(574.62 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present a software model and language for describing and programming the fine-grained aspects of interaction in a non-WIMP user interface, such as a virtual environment. Our approach is based on our view that the essence of a non-WIMP dialogue is a set of continuous relationships—most of which are temporary. The model combines a data-flow or constraint-like component for the continuous relationships with an event-based component for discrete interactions, which can enable or disable ...

Keywords: PMIW, interaction techniques, non-WIMP interface, specification language, state transition diagram, user interface management system (UIMS)


Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads:  Adobe Acrobat  QuickTime  Windows Media Player  Real Player**16 A methodology for testing spreadsheets**Gregg Rothermel, Margaret Burnett, Lixin Li, Christopher Dupuis, Andrei Sheretov
January 2001 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 10 Issue 1Full text available:  pdf(353.65 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Spreadsheet languages, which include commercial spreadsheets and various research systems, have had a substantial impact on end-user computing. Research shows, however, that spreadsheets often contain faults; thus, we would like to provide at least some of the benefits of formal testing methodologies to the creators of spreadsheets. This article presents a testing methodology that adapts data flow adequacy criteria and coverage monitoring to the task of testing spreadsheets. To accommodate ...

Keywords: software testing, spreadsheets**17 Incremental global reoptimization of programs**Lori L. Pollock, Mary Lou Soffa
April 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 14 Issue 2Full text available:  pdf(1.98 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Although optimizing compilers have been quite successful in producing excellent code, two factors that limit their usefulness are the accompanying long compilation times and the lack of good symbolic debuggers for optimized code. One approach to attaining faster recompilations is to reduce the redundant analysis that is performed for optimization in response to edits, and in particular, small maintenance changes, without affecting the quality of the generated code. Although modular program ...

Keywords: compiler optimization, incremental data flow analysis, incremental reoptimization, optimization dependencies**18 Fast address lookups using controlled prefix expansion**V. Srinivasan, G. Varghese
February 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 1Full text available:  pdf(259.60 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Internet (IP) address lookup is a major bottleneck in high-performance routers. IP address lookup is challenging because it requires a longest matching prefix lookup. It is compounded by increasing routing table sizes, increased traffic, higher-speed links, and the migration to 128-bit IPv6 addresses. We describe how IP lookups and updates can be made faster using a set of transformation techniques. Our main technique, controlled prefix expansion, transf ...

Keywords: Internet address lookup, binary search on levels, controlled prefix expansion, expanded tries, longest-prefix match, multibit tries, router performance**19 An incremental flow- and context-sensitive pointer aliasing analysis**Jyh-shiarn Yur, Barbara G. Ryder, William A. Landi
May 1999 **Proceedings of the 21st international conference on Software engineering**

PORTAL US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search:

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **software update revision signal** Found 51 of 142,346

Sort results by **relevance** Try an **Advanced Search**
 Display results **expanded form** Try this search in **The ACM Guide**
☐ Open results in a new window

Full text available: [pdf\(534.28 KB\)](#) [terms](#)

We describe a system that permits maintaining the software installed on several heterogeneous computers distributed over a network by taking advantage of the mobile agent paradigm. The applications are installed and updated only on the central server. When a new release of an application is installed on the server, agents are scattered along the network to update the application on the clients. To build a prototype system we use X-KLAIM, a programming language specifically designed to pr ...

Keywords: distributed software update, mobile agents, mobile code

4 **PLI workshops: A rule-based language for programming software updates**

Martin Erwig, Delin Ren
 December 2002 **ACM SIGPLAN Notices**, Volume 37 Issue 12
 Full text available: [pdf\(182.21 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe the design of a rule-based language for expressing changes to Haskell programs in a systematic and reliable way. The update language essentially offers update commands for all constructs of the object language (a subset of Haskell). The update language can be translated into a core calculus consisting of a small set of basic updates and update combinators. The key construct of the core calculus is a scope update mechanism that allows (and enforces) update specifications for the defin ...

Keywords: Haskell, type change, type correctness, update program, update safety

5 **A rule-based language for programming software updates**

Martin Erwig, Delin Ren
 October 2002 **Proceedings of the 2002 ACM SIGPLAN workshop on Rule-based programming**
 Full text available: [pdf\(118.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe the design of a rule-based language for expressing changes to Haskell programs in a systematic and reliable way. The update language essentially offers update commands for all constructs of the object language (a subset of Haskell). The update language can be translated into a core calculus consisting of a small set of basic updates and update combinators. The key construct of the core calculus is a scope update mechanism that allows (and enforces) update specifications for the defin ...

Keywords: type change, type correctness, update program, update safety

6 **Ergonomic standards for software: update for 1994**

Pat Billingsley
 April 1994 **Conference companion on Human factors in computing systems**
 Full text available: [pdf\(53.82 KB\)](#) Additional Information: [full citation](#)

7 **Wide-area monitoring of mobile objects: Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet**

Ting Liu, Christopher M. Sadler, Pei Zhang, Margaret Martonosi
 June 2004 **Proceedings of the 2nd international conference on Mobile systems, applications, and services**

September 2003 **Proceedings of the 31st annual ACM SIGUCCS conference on User services**

Full text available: [pdf\(211.45 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

This paper discusses the challenges of putting Apple Macintosh OSX into open access and computer lab environments.

Keywords: Macintosh, OSX, configuration, imaging, integration, lab, labs, maintenance, open access, security, software distribution, workstation

12 **Dynamic rebinding for marshalling and update, with destruct-time ?**

Gavin Bierman, Michael Hicks, Peter Sewell, Gareth Stoyke, Keith Wansborough
 August 2003 **ACM SIGPLAN Notices**, **Proceedings of the eighth ACM SIGPLAN international conference on Functional programming**, Volume 38 Issue 9
 Full text available: [pdf\(199.01 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Most programming languages adopt static binding, but for distributed programming an exclusive reliance on static binding is too restrictive: dynamic binding is required in various guises, for example when a marshalled value is received from the network, containing identifiers that must be rebound to local resources. Typically it is provided only by ad-hoc mechanisms that lack clean semantics. In this paper we adopt a foundational approach, developing core dynamic rebinding mechanisms as extension ...

Keywords: distributed programming, dynamic binding, dynamic update, lambda calculus, marshalling, programming languages, serialisation

13 **Recipe to lab management or the cookie cutter approach to building labs**

John H. Wilson
 November 2002 **Proceedings of the 30th annual ACM SIGUCCS conference on User services**
 Full text available: [pdf\(351.66 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

The ingredients in establishing and maintaining computer labs include determining the purpose and function of the lab; hardware and software needs; and infrastructure support. This paper outlines an approach to provide a recipe in establishing and maintaining labs. This approach attempts to lessen the guesswork and allows the lab manager to make more precise determinations when allocating resources. I have been managing computer labs for several years at Oklahoma State University and have develop ...

Keywords: budget, lab management, labor, materials

14 **A component and communication model for push systems**

Manfred Hauswirth, Mehdi Jazayeri
 October 1999 **ACM SIGSOFT Software Engineering Notes**, **Proceedings of the 7th European software engineering conference held jointly with the 7th ACM SIGSOFT international symposium on Foundations of software engineering**, Volume 24 Issue 6
 Full text available: [pdf\(1.50 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present a communication and component model for push systems. Surprisingly, despite the widespread use of many push services on the Internet, no such models exist. Our communication model contrasts push systems with client-server and event-based systems. Our component model provides a basis for comparison and evaluation of different push

1 **Dynamic software updating**

Michael Hicks, Jonathan T. Moore, Scott Nettles
 May 2001 **ACM SIGPLAN Notices**, **Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation**, Volume 36 Issue 5
 Full text available: [pdf\(1.44 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many important applications must run continuously and without interruption, yet must be changed to fix bugs or upgrade functionality. No prior general-purpose methodology for dynamic updating achieves a practical balance between flexibility, robustness, low overhead, and ease of use.

We present a new approach for C-like languages that provides type-safe dynamic updating of native code in an extremely flexible manner (code, data, and types may be updated, at programmer-determined times ...

2 **Impala: a middleware system for managing autonomic, parallel sensor systems**

Ting Liu, Margaret Martonosi
 June 2003 **ACM SIGPLAN Notices**, **Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 38 Issue 10
 Full text available: [pdf\(684.33 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Sensor networks are long-running computer systems with many sensing/compute nodes working to gather information about their environment, process and fuse that information, and in some cases, actuate control mechanisms in response. Like traditional parallel systems, communication between nodes is of fundamental importance, but is typically accomplished via wireless transceivers. One further key attribute of sensor networks is that they are almost always long running systems, intended to operate i ...

Keywords: middleware system, sensor networks, software adaptation, software update

3 **Agents, interactions, mobility and systems: Software update via mobile agent based programming**

Lorenzo Bettini, Rocco De Nicola, Michele Loreti
 March 2002 **Proceedings of the 2002 ACM symposium on Applied computing**
 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27172759&CFTOKEN=57625055>
 Results (page 1): "software update" "revision signal"

9/9/04<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27172759&CFTOKEN=57625055>
 Page 3 of 6Results (page 1): "software update" "revision signal"

9/9/04
 Page 4 of 6

Full text available: [pdf\(3.14 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

ZebraNet is a mobile, wireless sensor network in which nodes move throughout an environment working to gather and process information about their surroundings[10]. As in many sensor or wireless systems, nodes have critical resource constraints such as processing speed, memory size, and energy supply; they also face special hardware issues such as sensing device sample time, data storage/access restrictions, and wireless transceiver capabilities. This paper discusses and evaluates ZebraNet's syst ...

Keywords: event handling, middleware system, network communications, operation scheduling, sensor networks

8 **Software architecture based on communicating residential environments**

Erik Sandewall, Claes Strömberg, Henrik Sörensen
 March 1981 **Proceedings of the 5th international conference on Software engineering**
 Full text available: [pdf\(854.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes an alternative approach to software architecture, where the classical division of responsibilities between operating systems, programming languages and compilers, and so forth is revised. Our alternative is organized as a set of self-contained environments which are able to communicate pieces of software between them, and whose internal structure is predominantly descriptive and declarative. The base structure within each environment (its divers ...

9 **Applying data mining to software maintenance records**

Jelber Sayyad Shirabad, Timothy C. Lethbridge, Stan Matwin
 October 2003 **Proceedings of the 2003 conference of the Centre for Advanced Studies conference on Collaborative research**
 Full text available: [pdf\(140.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In a system maintained over a long time period, as is the case for legacy software, there will be many unknown and non-trivial relationships among components. Finding such hidden relationships may help software engineers in their maintenance activities. In this paper we present an approach whereby we mine software update records to find relationships between files that are changed together. The generalized models we present as results are obtained by using features extracted from different sourc ...

10 **DUX in practice II: Customer portal research and design**

Michael Kronthal
 June 2003 **Proceedings of the 2003 conference on Designing for user experiences**
 Full text available: [pdf\(105.28 KB\)](#) Additional Information: [full citation](#), [abstract](#)

The goal of this project is to improve the online experience of frequent and experienced users of PeopleSoft's customer extranet by designing and implementing a useful, usable, and satisfying Customer Portal. PeopleSoft.com's User Experience Specialist designed and executed a seven-week research program over a four-month period. The research process was split into a user-requirements gathering phase and a user-centered design phase. The project utilized a rigorous research approach ...

Keywords: anthropology, contextual interviews, experience design, experience strategy, focus groups, market research, portal, prototype testing, surveys, usability research, usability testing, usage study, user experience, user research, user-centered design

11 **Putting OSX in an open access lab: (or "The Joy of X")**

David L. R. Houston

systems and their design alternatives. We compare several prominent push systems using our component model. The component model consists of producers an ...

15 Organizing a campus computer coordinator group

Dennie Van Tassel

August 1990 **Proceedings of the 18th annual ACM SIGUCCS conference on User services**

Full text available: [pdf\(442.54 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

About three years ago the campus computer center, Computing and Telecommunications Services (CATS), started having meetings with campus computer coordinators. These computer coordinators support academic departments, administrative departments, and microcomputer laboratories. Once a month we all get together and recommend campus computing policy for the campus, which is then passed on to a high level Vice-Chancellor computing committee. A small part of the meeting is devoted to t ...

16 Mobile computing and applications (MCA): A declarative framework for adaptable applications in heterogeneous environments

P. Inverardi, F. Mancinelli, M. Nesti

March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

Full text available: [pdf\(242.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

In this paper we present an approach for developing adaptable software applications. The problem we are facing is that of a (possibly mobile) user who wants to download and execute an application from a remote server. The user's hosting device can be of different kinds (laptops, personal digital assistants, cellular phones, communicators, etc.) with specific hardware and software capabilities. The problem is to be able to decide whether the user's current device characteristics are compatible wl ...

17 Testbed directions and experience: Experience with an evolving overlay network testbed

David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, Robert Morris

July 2003 **ACM SIGCOMM Computer Communication Review**, Volume 33 Issue 3

Full text available: [pdf\(115.20 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

The MIT RON testbed consists of 36 Internet-connected nodes at 31 different sites. It has been in operation for two years. This paper presents an overview of the testbed, summarizes some of the research for which it has proved useful, and presents the lessons we learned during its development. The testbed has been useful both for our own research and for that of external researchers because of its heterogeneous, diverse network connections; its homogenous hardware and software platform; its inc ...

18 Migration: Optimizing the migration of virtual computers

Constantine P. Sapuntzakis, Ramesh Chandra, Ben Pfaff, Jim Chow, Monica S. Lam, Mendel Rosenblum

December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue 51

Full text available: [pdf\(1.68 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper shows how to quickly move the state of a running computer across a network, including the state in its disks, memory, CPU registers, and I/O devices. We call this state a *capsule*. Capsule state is hardware state, so it includes the entire operating system as well as applications and running processes. We have chosen to move x86 computer states because x86 computers are common, cheap, run the software we use, and have tools for migration. Unfortunately, x86 c ...

19 Caches and Memory Systems: Patchable instruction ROM architecture

Timothy Sherwood, Brad Calder

November 2001 **Proceedings of the international conference on Compilers, architecture, and synthesis for embedded systems**

Full text available: [pdf\(299.03 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Increased systems level integration has meant the movement of many traditionally off chip components onto a single chip including a processor, instruction storage, data path, and local memory. The design of these systems is driven by two conflicting goals, the need for reduced area and the need for rapid development times. The two current design options for instruction storage, ROM and Flash, are each highly optimized to one of these two goals but provide little compromise between them. ROM is u ...

20 Modules, abstract types, and distributed versioning

Peter Sewell

January 2001 **ACM SIGPLAN Notices , Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 36 Issue 3

Full text available: [pdf\(251.01 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In a wide-area distributed system it is often impractical to synchronise software updates, so one must deal with many coexisting versions. We study static typing support for modular wide-area programming, modelling separate compilation/linking and execution of programs that interact along typed channels. Interaction may involve communication of values of abstract types; we provide the developer with fine-grain versioning control of these types to support interoperation of old and new code. The s ...

Results 1 - 20 of 51

Result page: [1](#) [2](#) [3](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: [The ACM Digital Library](#) [The Guide](#)

US Patent & Trademark Office

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Modules, abstract types, and distributed versioning

Full text (252 KB)

Source Annual Symposium on Principles of Programming Languages archive
 Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, table of contents
 London, United Kingdom
 Pages: 236 - 247
 Year of Publication: 2001
 ISSN 0362-1340
 Also published in ...

Author Peter Sewell, Computer Laboratory, University of Cambridge

Sponsors SIGACT: ACM Special Interest Group on Algorithms and Computation Theory
 SIGPLAN: ACM Special Interest Group on Programming Languages

Publisher ACM Press, New York, NY, USA

 Additional Information: [abstract](#) [references](#) [citations](#) [index terms](#) [collaborative colleagues](#) [peer to peer](#)

Tools and Actions:

[Discussions](#) [Find similar Articles](#) [Review this Article](#)
[Save this Article to a Binder](#) [Display in BibTex Format](#)

DOI Bookmark:

 Use this link to bookmark this Article: <http://doi.acm.org/10.1145/360204.360225>
 What is a DOI?

ABSTRACT

In a wide-area distributed system it is often impractical to synchronise software updates, so one must deal with many coexisting versions. We study static typing support for modular wide-area programming, modelling separate compilation/linking and execution of programs that interact along typed channels. Interaction may involve communication of values of abstract types; we provide the developer with fine-grain versioning control of these types to support interoperability of old and new code. The system makes use of a second-class module system with singleton kinds; we give a novel operational semantics for separate compilation/linking and execution and prove soundness.

REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete list rather than only correct and linked references.

AWW95 Joe Armstrong, Robert Virding, Claes Wikström, Mike Williams, Concurrent programming in ERLANG (2nd ed.), Prentice Hall International (UK) Ltd., Hertfordshire, UK, 1996

BA99 Matthias Blume, Andrew W. Appel, Hierarchical modularity, ACM Transactions on Programming Languages and Systems (TOPLAS), v.21 n.4, p.813-847, July 1999

<http://portal.acm.org/citation.cfm?id=360225&coll=ACM&dl=ACM&CFID=27172759&CFTOKEN=57625...> 9/9/04
 Modules, abstract types, and distributed versioning

Page 3 of 5 Modules, abstract types, and distributed versioning

Page 4 of 5

and Computation, v.100 n.1, p.1-40, Sept. 1992

MTH90 Robin Milner, Mads Tofte, Robert Harper, The definition of Standard ML, MIT Press, Cambridge, MA, 1990

Ode95 Martin Odersky, Polarized Name Passing, Proceedings of the 15th Conference on Foundations of Software Technology and Theoretical Computer Science, p.324-337, December 18-20, 1995

PS96 Benjamin Pierce and Davide Sangiorgi, Typing and subtyping for mobile processes, Mathematical Structures in Computer Science, 6(5):409-454, 1996.

PT00 Benjamin C. Pierce, David N. Turner, Pict: a programming language based on the π -Calculus, Proof, language, and interaction: essays in honour of Robin Milner, MIT Press, Cambridge, MA, 2000

RFS98 Claudio V. Russo, Types For Modules, PhD thesis, University of Edinburgh, LFCS, 1998. ECS-LFCS-98-389.

Sew00 Peter Sewell, Modules, abstract types, and distributed versioning. Technical Report 506, University of Cambridge, 2000. Available from <http://www.cl.cam.ac.uk/users/pes20/>.

SWP98 Peter Sewell, Pawel T. Wojciechowski, and Benjamin C. Pierce, Location independence for mobile agents. In Proceedings of the Workshop on Internet Programming Languages (Chicago), May 1998. Full version appeared in LNCS 1686.

US01 Asis Unyapoth, Peter Sewell, Nomadic pict: correct communication infrastructure for mobile computation, Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.116-127, January 2001, London, United Kingdom

Woj00 Pawel T. Wojciechowski, Nomadic Pict: Language and Infrastructure Design for Mobile Computation, PhD thesis, Computer Laboratory, University of Cambridge, June 2000. Available as Technical Report 492.

WS00 Pawel T. Wojciechowski, Peter Sewell, Nomadic Pict: Language and Infrastructure Design for Mobile Agents, IEEE Concurrency, v.8 n.2, p.42-52, April 2000

ZGM99 Steve Zdancewicz, Dan Grossman, Greg Morrisett, Principals in programming languages: a syntactic proof technique, Proceedings of the fourth ACM SIGPLAN International conference on Functional programming, p.197-207, September 27-29, 1999, Paris, France

CITINGS 4

James J. Leifer, Gilles Peskine, Peter Sewell, Keith Wansbrough, Global abstraction: safe marshalling with hash types, ACM SIGPLAN Notices, v.38 n.9, p.87-98, September 2003

Eijiro Sumii, Benjamin C. Pierce, A bisimulation for dynamic sealing, ACM SIGPLAN Notices, v.39 n.1, p.161-172, January 2004

Andreas Rossberg, Generativity and dynamic opacity for abstract types, Proceedings of the 5th ACM SIGPLAN International conference on Principles and practice of declarative programming, p.241-252, August 27-29, 2003, Uppsala, Sweden

Dominic Duggan, Type-based hot swapping of running modules (extended abstract), ACM SIGPLAN Notices, v.36 n.10, October 2001

Car97 Luca Cardelli, Program fragments, linking, and modularization, Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.266-277, January 15-17, 1997, Paris, France

CDG+89 Luca Cardelli, James Donahue, Lucille Glassman, Mick Jordan, Bill Kalsow, and Greg Nelson, Modula-3 report (revised), Technical report, DEC SRC, September 1989. SRC-052.

CL90 L. Cardelli and X. Leroy, Abstract types and the dot notation, In Programming Concepts and Methods, IFIP State of the Art Reports, pages 479-504. North Holland, March 1990. Also appeared as SRC Research Report 56.

Dro00 Sophia Drossopoulou, An Abstract Model of Java Dynamic Linking and Loading, Selected papers from the Third International Workshop on Types in Compilation, p.53-84, September 21, 2000

Dug00 Dominic Duggan, Sharing in Typed Module Assembly Language, Selected papers from the Third International Workshop on Types in Compilation, p.85-116, September 21, 2000

FF98 Matthew Flatt, Matthias Felleisen, Units: cool modules for HOT languages, Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation, p.236-248, June 17-19, 1998, Montreal, Quebec, Canada

FGL+96 Cédric Fournet, Georges Gonthier, Jean-Jacques Lévy, Luc Maranget, Didier Rémy, A Calculus of Mobile Agents, Proceedings of the 7th International Conference on Concurrency Theory, p.406-421, August 26-29, 1996

GM99 Neal Glew, Greg Morrisett, Type-safe linking and modular assembly language, Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.250-261, January 20-22, 1999, San Antonio, Texas, United States

HL94 Robert Harper, Mark Lillibridge, A type-theoretic approach to higher-order modules with sharing, Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.123-137, January 16-19, 1994, Portland, Oregon, United States

HN00 Michael W. Hicks, Scott Nettles, Active Networking Means Evolution (or Enhanced Extensibility Required), Proceedings of the Second International Working Conference on Active Networks, p.16-32, October 16-18, 2000

HWC00 Michael W. Hicks, Stephanie Weirich, Karl Cray, Safe and Flexible Dynamic Linking of Native Code, Selected papers from the Third International Workshop on Types in Compilation, p.147-176, September 21, 2000

Ler94 Xavier Leroy, Manifest types, modules, and separate compilation, Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.109-122, January 16-19, 1994, Portland, Oregon, United States

Ler96 Xavier Leroy, A syntactic theory of type generativity and sharing, Journal of Functional Programming, 6(5):667-698, 1996.

Ler00 Xavier Leroy, A modular module system, Journal of Functional Programming, 10(3):269-303, May 2000.

LI97 Mark Lillibridge, Translucent Sums: A Foundation for Higher-Order Module Systems, PhD thesis, CMU, 1997. CMU-CS-97-122.

MPW92 Robin Milner, Joachim Parrow, David Walker, A calculus of mobile processes, I, Information

INDEX TERMS

Primary Classification:

D. Software

D.3 PROGRAMMING LANGUAGES

D.3.3 Language Constructs and Features

Subjects: Modules, packages

Additional Classification:

D. Software

D.1 PROGRAMMING TECHNIQUES

D.3 PROGRAMMING LANGUAGES

D.3.3 Language Constructs and Features

Subjects: Abstract data types

K. Computing Milieux

K.6 MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS

K.6.3 Software Management

Subjects: Software development

General Terms:

Design, Languages, Performance, Theory

Collaborative Colleagues:

Peter Sewell:	Gavin Bierman	Jan Vitek
	Gian Luca Cattani	Keith Wansbrough
	Michael Hicks	Pawel T. Wojciechowski
	James J. Leifer	
	Michael Norrish	
	Gilles Peskine	
	Benjamin C. Pierce	
	Andrei Serjantov	
	Gareth Stoyke	
	Asis Unyapoth	

Peer to Peer - Readers of this Article have also read:

- Constructing reality
 Proceedings of the 11th annual international conference on Systems documentation
 Douglas A. Powell, Norman R. Ball, Mansel W. Griffiths
- Toward a real-time Ada design methodology
 Proceedings of the conference on TRI-ADA '90
 Norman R. Howes
- Fashioning conceptual constructs in Ada
 Proceedings of the conference on TRI-ADA '90
 Robert C. Shock

- Reuse: the two concurrent life cycles paradigm
Proceedings of the conference on TRI-ADA '90
Richard Drake , William Ett
- An experiment with Graphite
Proceedings of the conference on TRI-ADA '90
John Chludzinski , Robert Chi Tau Lai

↑ **This Article has also been published in:**

- **ACM SIGPLAN Notices**
Volume 30 , Issue 3 (March 2001)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☐ The ACM Digital Library ☐ The Guide

stored revision identity

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **stored revision identity**

Found 15,684 of 142,346

 Sort results by [Save results to a Binder](#) [Try an Advanced Search](#)
 Display results [Search Tips](#) [Try this search in The ACM Guide](#)
☐ Open results in a new window

 Results 1 - 20 of 200 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)
 Best 200 shown Relevance scale ☐ ☐ ☐ ☐

- 1 **PocketLens: Toward a personal recommender system**
 Bradley N. Miller, Joseph A. Konstan, John Riedl
 July 2004 **ACM Transactions on Information Systems (TOIS)**, Volume 22 Issue 3
 Full text available: [pdf\(1.10 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)
- Recommender systems using collaborative filtering are a popular technique for reducing information overload and finding products to purchase. One limitation of current recommenders is that they are not portable. They can only run on large computers connected to the Internet. A second limitation is that they require the user to trust the owner of the recommender with personal preference data. Personal recommenders hold the promise of delivering high quality recommendations on palmtop computers, e ...

Keywords: Collaborative Filtering, Peer-to-Peer Networking, Privacy, Recommender Systems

- 2 **Production of large computer programs**
 H. D. Benington
 March 1987 **Proceedings of the 9th international conference on Software Engineering**
 Full text available: [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)
- The paper is adapted from a presentation at a symposium on advanced programming methods for digital computers sponsored by the Navy Mathematical Computing Advisory Panel and the Office of Naval Research in June 1956. The author describes the techniques used to produce the programs for the Semi-Automatic Ground Environment (SAGE) system.

- 3 **Bitslice: representation without taxation**
 Robert Bernecky, Peter Wooster
 June 2003 **Proceedings of the 2003 conference on APL: stretching the mind**
 Full text available: [pdf\(105.97 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)
- We describe bitslice, a high-performance utility for parallel bit-level indexing of array data. We claim that our results demonstrate the importance of bit arrays as a fundamental data type and of being able to view array data at the Boolean array level. We present performance figures that confirm our claims, taken from a large data broadcast application.

- 4 **IT implementation through the lens of organizational learning: a case study of insuror**

<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27174275&CFTOKEN=18476005>
 Results (page 1): stored revision identity

 Full text available: [pdf\(1.10 MB\)](#) [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A Shared Book helps the users of an office information system create a multiple-part publication and manage it throughout its life cycle. The Shared Book supports simultaneous collaboration both by allowing different workers to work on different parts at the same time and by ensuring that workers use the current revision of each part. It protects publication information by providing locking and access control. The Shared Book communicates the publication's current state to ...

- 9 **Persistent storage for a workflow tool implemented in Smalltalk**
 Bob Beck, Steve Hartley
 October 1994 **ACM SIGPLAN Notices , Proceedings of the ninth annual conference on Object-oriented programming systems, language, and applications**, Volume 29 Issue 10
 Full text available: [pdf\(1.74 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)
- This paper describes a new workflow model and its implementation in Smalltalk. The paper also details problems with using a RDBMS as the persistent store for the workflow tool and the subsequent experiences in using an ODBMS for this purpose. The final solution was a coexistence approach, using the RDBMS for legacy corporate data and the ODBMS for the process description and workflow status data.

- 10 **Posters: Dynamic personal roles for ubiquitous computing**
 Robert E. McGrath, M. Dennis Mickunas
 October 2003 **Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**
 Full text available: [pdf\(1.31 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)
- This paper presents doctoral research on a key problem for ubiquitous computing: implementation of representatives for physical objects, particularly people. This poster outlines an approach to implementing dynamic personal roles suitable for a ubiquitous computing environment.
- Keywords:** post-object programming, roles, ubiquitous computing

- 11 **A Status Report on Computing Algorithms for Mathematical Programming**
 William W. White
 September 1973 **ACM Computing Surveys (CSUR)**, Volume 5 Issue 3
 Full text available: [pdf\(3.02 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

- 12 **Configuration control for evolutionary software products**
 Osamu Shigo, Yoshio Wada, Yulchi Terashima, Kanji Iwamoto, Takashi Nishimura
 September 1982 **Proceedings of the 6th international conference on Software engineering**
 Full text available: [pdf\(800.05 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
- This paper describes the concept of and a system for configuration control for evolutionary software products. In which a wide spectrum of varied software products are being continuously evolved, along with rapid advancements in hardware technologies. The system contains a database for dealing with the overall configuration structure, including hierarchical product structure with change status, master file directories, difficulty occurrences and user information. The data representing the co ...

 Kwang-Tat Ang, James Y. L. Thong, Chee-Sing Yap
 December 1997 **Proceedings of the eighteenth international conference on Information systems**
 Full text available: [pdf\(95.41 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#)
Keywords: IT Implementation, case study research, double-loop learning, organizational learning

- 5 **A database model for effective configuration management in the programming environment**
 Karen E. Huff
 March 1981 **Proceedings of the 5th international conference on Software engineering**
 Full text available: [pdf\(747.85 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The effective management of configurations by programmers requires automatic techniques which are operative in the program development environment. In this paper, an abstract model is developed to cover the significant aspects of a typical programming environment pertinent to configuration management, using a database to capture configuration knowledge. The two aspects of the model deal with configuration identification and configuration control. In considering configuration identification, ...

- 6 **The revised ARPANET routing metric**
 A. Khanna, J. Zinky
 August 1989 **ACM SIGCOMM Computer Communication Review , Symposium proceedings on Communications architectures & protocols**, Volume 19 Issue 4
 Full text available: [pdf\(1.30 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The ARPANET routing metric was revised in July 1987, resulting in substantial performance improvements, especially in terms of user delay and effective network capacity. These revisions only affect the individual link costs (or metrics) on which the PSN (packet switching node) bases its routing decisions. They do not affect the SPF ("shortest path first") algorithm employed to compute routes (installed in May 1979). The previous link metric was packet delay averaged over a ten s ...

- 7 **Contexts—a partitioning concept for hypertext**
 Norman M. Delisle, Mayer D. Schwartz
 April 1987 **ACM Transactions on Information Systems (TOIS)**, Volume 5 Issue 2
 Full text available: [pdf\(1.49 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Hypertext systems provide good information management support for a wide variety of documentation efforts. These efforts range from developing software to writing a book. However, existing hypertext systems provide poor support for collaboration among teams of authors. This paper starts by briefly describing properties of several existing hypertext systems. Then several models for forming partitions in a hypertext database are examined and contexts, a partitioning scheme that supports multi ...

- 8 **Shared books: collaborative publication management for an office information system**
 Brian T. Lewis, Jeffrey D. Hodges
 April 1988 **ACM SIGOIS Bulletin , Conference Sponsored by ACM SIGOIS and IEEECS TC-OA on Office Information systems**, Volume 9 Issue 2-3
 Additional Information:

[9/9/04http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27174275&CFTOKEN=18476005](http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27174275&CFTOKEN=18476005)
 Page 3 of 5Results (page 1): stored revision identity

[9/9/04](http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=27174275&CFTOKEN=18476005)
 Page 4 of 5

- 13 **TransformGen: automating the maintenance of structure-oriented environments**
 David Garland, Charles W. Krueger, Barbara Staudt Lerner
 May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 3
 Full text available: [pdf\(3.10 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A serious problem for programs that use persistent data is that information created and maintained by the program becomes invalid if the persistent types used in the program are modified in a new release. Unfortunately, there has been little systematic treatment of the problem; current approaches are manual, ad hoc, and time consuming both for programmers and users. In this article we present a new approach. Focusing on the special case of managing abstract syntax trees in structure-oriented ...

Keywords: schema evolution, structure-oriented environments, type evolution

- 14 **Revised report on the algorithmic language scheme**
 J Rees, W Clinger
 December 1986 **ACM SIGPLAN Notices**, Volume 21 Issue 12
 Full text available: [pdf\(4.06 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

- 15 **Design of the Mnome persistent object store**
 J. Elliot B. Moss
 April 1990 **ACM Transactions on Information Systems (TOIS)**, Volume 8 Issue 2
 Full text available: [pdf\(3.22 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Mnome project is an investigation of techniques for integrating programming language and database features to provide better support for cooperative, information-intensive tasks such as computer-aided software engineering. The project strategy is to implement efficient, distributed, persistent programming languages. We report here on the Mnome persistent object store, a fundamental component of the project, discussing its design and initial prototype. Mnome stores objects

- 16 **Numerical applications: updating the product form of the inverse for the revised simplex method**
 G. B. Dantzig, R. P. Harvey, R. D. McKnight
 August 1965 **Proceedings of the 1965 20th national conference**
 Full text available: [pdf\(311.02 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

COMPUTER CODES for solving linear programs by the simplex method usually use one of three forms in representing the problem during the course of solution. These are: (a) - the standard form or original simplex method; (b) - the revised simplex method with explicit inverse; and (c) - the revised simplex method with inverse in product form. [For a comparison of the relative efficiencies of the three methods, see text by Wolfe and Cutler]

- 17 **Anonymous credit cards and their collusion analysis**
 Steven H. Low, Nicholas F. Maxemchuk, Sanjoy Paul
 December 1996 **IEEE/ACM Transactions on Networking (TON)**, Volume 4 Issue 6
 Full text available: [pdf\(860.04 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

- 18 **Hypertext versioning: The molhado hypertext versioning system**
 Tien N. Nguyen, Ethan V. Munson, John T. Boyland

August 2004 **Proceedings of the fifteenth ACM conference on Hypertext & hypermedia**

Full text available:  pdf(943.39 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes *Molhado*, a hypertext versioning and software configuration management system that is distinguished from previous systems by its flexible product versioning and structural configuration management model. The model enables a unified versioning framework for atomic and composite software artifacts, and hypermedia structures among them in a fine-grained manner at the logical level. Hypermedia structures are managed separately from documents' contents. Molhado explicitly r ...

Keywords: hypertext versioning, software configuration management, software engineering, version control

19 A future APL: examples and problems

M. Gfeffer

July 1989 **ACM SIGAPL APL Quote Quad , Conference proceedings on APL as a tool of thought**, Volume 19 Issue 4

Full text available:  pdf(861.59 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

AIDA is a modern APL featuring operator-less syntax, function attributes, function arrays, array parts and a hierarchical object library. The above areas are illustrated by examples on a model interpreter written in SHARP APL®. The relation of the AIDA extensions to APL application systems design is discussed and some problem areas requiring further thought are identified.

20 Circuit diagram generation via functional logic

S. M. Shenkman

June 1973 **Proceedings of the 10th workshop on Design automation**

Full text available:  pdf(346.37 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

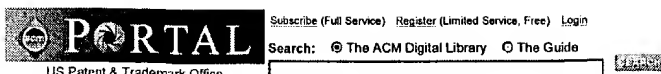
A system used by Gilbert Associates, Inc. to produce, via computer, electrical elementary wiring diagrams directly from Boolean expressions is described here. Discussion is concentrated on the aspect of using a functional input format to create circuits for control of power plant systems and equipment. Information is processed on a single medium of communication for all responsible participants. A "Functional Diagram" contains information from and for system, control, instrument ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  Adobe Acrobat  QuickTime  Windows Media Player  Real Player



THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Configuration control for evolutionary software products

Full text Pdf (600 KB)

Source [International Conference on Software Engineering archive](#)
[Proceedings of the 6th international conference on Software engineering](#) [table of contents](#)
 Tokyo, Japan
 Pages: 68 - 75
 Year of Publication: 1982
 ISSN: 0270-5257

Authors [Osamu Shigo](#)
[Yoshio Wada](#)
[Yuichi Terashima](#)
[Kanji Iwamoto](#)
[Takashi Nishimura](#)

Sponsors [IPSI : Information Processing Society of Japan](#)
[NBS : National Bureau of Standards](#)
[IEEE-CS : Computer Society](#)
[SIGSOFT : ACM Special Interest Group on Software Engineering](#)

Publisher [IEEE Computer Society Press](#) [Los Alamitos, CA, USA](#)

[Additional Information: abstract](#) [references](#) [citations](#) [index terms](#) [collaborative colleagues](#) [peer to peer](#)

Tools and Actions: [Discussions](#) [Find similar Articles](#) [Review this Article](#)
[Save this Article to a Binder](#) [Display in BibTex Format](#)

↑ ABSTRACT

This paper describes the concept of and a system for configuration control for evolutionary software products, in which a wide spectrum of varied software products are being continuously evolved, along with rapid advancements in hardware technologies. The system contains a database for dealing with the overall configuration structure, including hierarchical product structure with change status, master file directories, difficulty occurrences and user information. The data representing the configuration can be in more abstract or macroscopic level than the traditional software configuration control, since the configuration manager should control the overall outlines for all software products. The concept and system described in the paper have been used for intelligent terminal software product management in NEC's Data Terminals Division.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

- 1 W. Bryan, C. Chadbourn and S. Siegel: "Tutorial: Software Configuration Management", IEEE Computer Society Press, 1980.

- 2 M. G. Spiegel: "Prototyping: An Approach to Information and Communication System Design", ACM SIGMETRICS Performance Evaluation Review, Vol.10, No.1, pp.9-19, Spring 1981.

- 3 M. Dyer: "The Management of Software Engineering, Part IV: Software Development Practices", IBM Systems Journal, Vol.19, No.4, pp.451-465, April 1980.

- 4 T. Gilb: "Evolutionary Development", Software Engineering Notes, Vol.6, No.2, p.17, April 1981.

- 5 Walter F. Tichy, Software development control based on module interconnection, Proceedings of the 4th international conference on Software engineering, p.29-41, September 17-19, 1979, Munich, Germany

- 6 Bennett P. Lientz, E. Burton Swanson, Software Maintenance Management, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1980

- 7 K. Iwamoto, S. Fujibayashi, O. Shigo and T. Nishimura: "SDMS: Software Development and Maintenance Support System", 20th Annual conference of Information Processing Society of Japan, pp.327-328, 1979 (in Japanese).

↑ CITINGS

C. V. Ramamoorthy, Yutaka Usuda, Atul Prakash, W. T. Tsai, The Evolution Support Environment System, IEEE Transactions on Software Engineering, v.16 n.11, p.1225-1234, November 1990

↑ INDEX TERMS

Primary Classification:

D. Software
 ↪ D.2 SOFTWARE ENGINEERING
 ↪ D.2.2 Design Tools and Techniques
 ↪ **Subjects:** Evolutionary prototyping

Additional Classification:

D. Software
 ↪ D.2 SOFTWARE ENGINEERING
 ↪ D.2.9 Management
 ↪ **Subjects:** Productivity

K. Computing Milieux
 ↪ K.6 MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS
 ↪ K.6.3 Software Management
 ↪ **Subjects:** Software development

General Terms:
 Management, Theory

↑ Collaborative Colleagues:

<http://portal.acm.org/citation.cfm?id=807749&coll=ACM&dl=ACM&CFID=27174275&CFTOKEN=18476...> 9/9/04 <http://portal.acm.org/citation.cfm?id=807749&coll=ACM&dl=ACM&CFID=27174275&CFTOKEN=18476...> 9/9/04
 Configuration control for evolutionary software products Page 3 of 4 Configuration control for evolutionary software products Page 4 of 4

Kanji Iwamoto: [Takashi Nishimura](#)
[Osamu Shigo](#)
[Yuichi Terashima](#)
[Yoshio Wada](#)

Takashi Nishimura: [Michita Imai](#) [Gen Suzuki](#)
[Kanji Iwamoto](#) [Tim Takeuchi](#)
[Yuichi Kato](#) [Tomoaki Tanaka](#)
[Yasuyuki Kiyosue](#) [Yuichi Terashima](#)
[Carol Leick](#) [Yoshio Wada](#)
[Norihiko Matsuura](#) [Takashi Yamana](#)
[Keiichi Sasaki](#)
[Osamu Shigo](#)
[Kazunori Shimamura](#)
[Shobei Sugawara](#)

Osamu Shigo: [Kanji Iwamoto](#)
[Takashi Nishimura](#)
[Yuichi Terashima](#)
[Yoshio Wada](#)

Yuichi Terashima: [Kanji Iwamoto](#)
[Takashi Nishimura](#)
[Osamu Shigo](#)
[Yoshio Wada](#)

Yoshio Wada: [Kanji Iwamoto](#)
[Takashi Nishimura](#)
[Osamu Shigo](#)
[Yuichi Terashima](#)

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

↑ Peer to Peer - Readers of this Article have also read:

- [HyperWeb: a framework for hypermedia based environments](#)
ACM SIGSOFT Software Engineering Notes 17, 5
 James C. Ferrans, David W. Hurst, Michael A. Sennett, Burton M. Covnot, Wenguang Ji, Peter Kaja, Wei Ouyang
- [Real theorem provers deserve real user interfaces](#)
ACM SIGSOFT Software Engineering Notes 17, 5
 Laurent Théry, Yves Bertot, Gilles Kahn
- [Building Integrated software development environments. Part I: tool specification](#)
ACM Transactions on Software Engineering and Methodology (TOSEM) 1, 2
 G. Engels, C. Lewerentz, M. Nagl, W. Schäfer, A. Schörr
- [An analysis technique for examining integration in a project support environment](#)
ACM SIGSOFT Software Engineering Notes 17, 5
 Alan W. Brown, Peter H. Feiler
- [Software engineering programs at the Wang Institute of Graduate Studies](#)
Proceedings of the 1983 annual conference on Computers : Extending the human resource
 Richard E. Fairley, Nancy Martin